

REMARKS

In the Office Action, the Examiner indicated that claims 1, 4-6 and 8-10 are pending in the application and the Examiner rejected all of the claims.

Claim Objections

On page 2 of the Office Action, the Examiner objected to claims 9 and 10 for being unclear as to their dependency. Applicant submits that each of these claims are claims drafted using the well-known and accepted "product by process" claiming method and are thus proper claims, each depending from claim 1. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the objections to claims 9 and 10.

Rejection under 35 U.S.C. §103

On page 2 of the Office Action, the Examiner rejected claims 1, 4-6 and 8-10 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,874,148 to Richardson in view of IBM Technical Disclosure ("Forwarder Dynamic Link Libraries as a Method for Servicing Software," IBM Technical Disclosure Bulletin, IBM Corp., New York, US, vol. 38, no. 11, November 1, 1995, pps. 407-408, XP000547408 ISSN: 0018-, hereinafter "IBM").

The Present Invention

The present invention discloses a dynamic link library (DLL) in a computing device provided in the form of a first part and an extension part. The first part has selected entry point

ordinals by which an application program may link to first functions. The application program may only link to further functions via the extension part of the DLL.

The Examiner Has Not Established a Prima Facie Case of Obviousness

KSR (*KSR International Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 82 USPQ2d 1385

(2007) requires that an Examiner provide “some articulated reasoning with some rationale underpinning to support the legal conclusion of obviousness.” Further, an Examiner must “identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does,” In addition, the Examiner must make “explicit” this rationale of “the apparent reason to combine the known elements in the fashion claimed,” including a detailed explanation of “the effects of demands known to the design community or present in the marketplace” and “the background knowledge possessed by a person having ordinary skill in the art.”

Applicant respectfully disagrees with the Examiner’s conclusion that the invention would have been obvious to a skilled person. The combination of Richardson and the IBM document does not disclose every feature of the claimed invention, and they do not, taken alone or in combination, provide any reason or motivation for the skilled person to arrive at the claimed invention.

More specifically, Richardson discloses the idea of providing certain graphical functions as code in a shared library that can be accessed by multiple programs (see first three sentences of Abstract). A user chooses which graphical programs to place in the library (or “export to” the library). The code in the library can then be accessed by another program by means of a call into

the library. Applicant notes that this is a very common way of arranging functions in a computing device, and is acknowledged as prior art in the present application at page 3, first and second full paragraphs.

The Richardson invention aims to solve prior art problems of exporting graphical functions, which include the need for a process switch when calling an exported function (see column 3, lines 53-60). It achieves this by providing that the user, having chosen a graphical program to export, then decides which part of the graphical program should act as the starting point of the function in the library, and what other functionality is required before the graphical program can be executed – in other words, he specifies the desired functional interface for the shared library entry point (see Figure 3 and middle portion of Abstract). The functional interface handles issues such as inputs and outputs of graphical data types, and may translate parameters passed by a program into a form that is expected by the exported graphical program (see last sentence of Abstract).

The portions of Richardson specifically cited by the Examiner simply discuss the outline of the Richardson invention as summarized above. In Richardson, there is no mention of how the exported programs in the shared library are identified by the program wishing to call those programs. In contrast, in the present claims it is clearly recited that *ordinal numbers* (i.e. position identifiers) are used to link executable programs to the functions in the library. As is explained in the description of the present application, names of functions are commonly used as an alternative method of linking executable programs to the functions in the library. The problems that are solved by the presently claimed invention are specific to ordinal linking arrangements – these problems were referred to in the applicant's previous response dated 27 May 2008, and are

described in detail in the present application at page 4, final two paragraphs and at page 6, second paragraph to page 9, second paragraph. In contrast, the problem that Richardson attempts to solve is unrelated to the type of linking arrangement used, since Richardson makes no reference to either type of linking. Applicant therefore contests the Examiner's assertion that Richardson discloses the feature of "causing the executable program to link to functions in the first part directly by means of the associated ordinal numbers" as is claimed herein. Nowhere in the Richardson document are ordinal numbers even mentioned.

In the IBM document, a library architecture is disclosed in which every function in the library (the primary DLL) is linked indirectly via a forwarder DLL. The aim of this invention is to enable updates to shipped software, and the updates are effected through the forwarder DLLs. Specifically, functions in the forwarder DLLs can be replaced with corrective functions (see page 407, third paragraph). The primary DLL itself is intended *never to be amended* (see page 408, final paragraph). This is in direct contrast to the aim of the present invention, which is precisely *to facilitate the amendment* of the underlying (or primary) DLL, without causing the calling executable program to call an incorrect ordinal number. The IBM disclosure does not provide any solution to the ordinal number problems addressed in the present application and, in fact, teaches away from it.

In summary, there is no disclosure in either Richardson or the IBM document of the claimed features that ordinal numbers are used for linking to functions in a library; nor is there any disclosure of the claimed features of splitting a DLL into two parts and providing different access mechanisms for each part, both of which are novel and non-obvious concepts. Therefore a combination of Richardson and the IBM document does not disclose every feature of the claimed

invention. Moreover, neither document provides any motivation for either linking by ordinal or using a novel arrangement involving the two parts of the library as claimed in the present claims.

The problems which the present invention overcomes are not mentioned or even alluded to in either of the cited prior art documents, and thus the skilled person is provided with no indication that there could be any advantage in providing a combination of directly linked and indirectly linked functions within the same DLL.

The motivation which the Examiner suggests (in point 9 of the office action), for combining the Richardson disclosure with the concept of indirect linking from the IBM document, is that this would improve “the integrity of Richardson’s system by allowing the software developers with no access to the underlying primary code to properly design, document interfaces, fix, and enhance in the DLLs”. Although this may be true, if the skilled person wished to achieve this aim and was prompted to modify Richardson in order to achieve it, then he would provide *every* function in Richardson’s primary code with a forwarder DLL, as suggested in IBM. He would have no reason whatsoever for leaving some functions to be linked to directly by the executable program – doing so would mean that no changes could be made to those functions, and thus the advantages of the IBM document would not be achieved. Thus, a combination of Richardson and the IBM document would not lead the skilled person to arrive at the present invention, which involves allowing direct access to *some* functions while forcing indirect access for the remaining functions, in order to solve a problem which is not considered by either of the prior art documents.

Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection of claims 1, 4-6 and 8-10 under 35 USC §103.

Conclusion

The present invention is not taught or suggested by the prior art. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection of the claims. An early Notice of Allowance is earnestly solicited.

The Commissioner is hereby authorized to charge any fees associated with this communication to Deposit Account No. 50-4364.

Respectfully submitted

December 1, 2008
Date

/Mark D. Simpson/
Mark D. Simpson, Esquire
Registration No. 32,942

SAUL EWING LLP
Centre Square West
1500 Market Street, 38th Floor
Philadelphia, PA 19102-2189
Telephone: 215 972 7880
Facsimile: 215 972 4169
Email: MSimpson@saul.com